

UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS

(Universidad del Perú, DECANA DE AMÉRICA)

FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

ESCUELA PROFESIONAL DE INGENIERíA DE SISTEMAS

**SÍLABO**

1. **INFORMACIÓN GENERAL**

|  |  |  |  |
| --- | --- | --- | --- |
| 1.1 | Nombre de la asignatura | : | **DISEÑO Y ANALISIS DE ALGORITMOS** – **Plan 2018** |
| 1.2 | Código de la asignatura | : | 2010502 |
| 1.3 | Tipo de Asignatura | : | Obligatorio |
| 1.4 | Área de Estudios | : | Específica |
| 1.5 | Numero de Semanas | : | 16 |
| 1.6 | Horas semanales | : | Teoría: 2, Laboratorio: 2 |
| 1.7 | Semestre Académico | : | 2024-1 |
| 1.8 | Ciclo | : | V |
| 1.9 | Créditos | : | 3 |
| 1.10 | Modalidad | : | Presencial |
| 1.11 | Pre-requisito | : | ALGORITMICA Y PROGRAMACION ORIENTADA A OBJETOS |
| 1.12 | Docente(s) | : | Cortez Vásquez Augusto Parcemon (acortezv@unmsm.edu.pe)  Guerra Grados, Luis Angel (lguerrag1@unmsm.edu.pe) |

1. **SUMILLA:**

Esta asignatura pertenece al área de formación específica, es de naturaleza teórico y práctico, tiene el propósito que el estudiante maneje la eficiencia de los aIgoritmos orientados a objetos. Los contenidos principales de la asignatura son: Formalismo y abstracción en el diseno de algoritmos, análisis de la eficiencia de algoritmos, complejidad temporal, complejidad espacial, complejidad asintótica. Recursión y resolución de recurrencias, la técnica divide y vencerás, métodos de clasificación, métodos de dispersión nueva, algoritmos de retroceso, algoritmos voraces, programación dinámica, ramificación y poda, y programación paralela, computabiIidad y complejidad algorítmica.

1. **COMPETENCIAS DEL PERFIL DE EGRESO A LA QUE CONTRIBUYE LA ASIGNATURA**

|  |  |  |  |
| --- | --- | --- | --- |
| **Código** | **Descripción** | **Tipo** | **Nivel** |
| CE1.2 | **Analiza la complejidad algorítmica**  Realiza análisis de complejidad algorítmica basado en metodologías, estándares y métricas. | Específica | Intermedio |
| CE2.2 | **Identifica, evalúa soluciones**  Identifica, evalúa soluciones de problemas basado en el resultado del análisis y eficiencia de los algoritmos. | Específica | Intermedio |
| CT1 | **Aplica metodologías, estándares**  Aplica las metodologías, estándares para el análisis de algoritmos en el desarrollo de problemas, sistemas de información, con compromiso ético y solidario. | Técnica | Intermedio |

1. **LOGROS DE APRENDIZAJE (Competencia de la asignatura)**

Al finalizar la asignatura el estudiante realiza el análisis de complejidad algorítmica para determinar la eficiencia de un algoritmo utilizando metodologías, estándares de manera ética y responsable.

1. **LOGROS POR UNIDAD (Capacidades)**

* **Unidad 1.** **Análisis de algoritmos**: Utiliza los fundamentos matemáticos (series, sumatorias, logaritmos, límites, derivadas), notaciones asintóticas, métodos para el análisis de algoritmos y medir la eficiencia de los algoritmos, tanto estructurados como recursivos, con actitud innovadora y responsable.
* **Unidad 2.** **Análisis de algoritmos de ordenación, búsqueda y grafos. Dispersión:** Aplica análisis de complejidad en algoritmos de ordenación, búsqueda, grafos mediante algoritmos para ordenar, buscar, algoritmos en grafos, algoritmos de búsqueda por dispersión, con actitud ética y responsable.
* **Unidad 3. Técnicas de diseño de algoritmos:** Comprende las diversas técnicas de diseño de algoritmos, sus ventajas y desventajas, análisis de complejidad, para la solución de problemas de distinta naturaleza, principalmente los de optimización, utilizando los algoritmos voraces, división y conquista, programación dinámica, algoritmos de retroceso, problemas P, NP, con actitud de innovación y responsabilidad.

1. **PROGRAMACIÓN DE CONTENIDOS**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Unidad I: Análisis de algoritmos** | | | | |
| **Capacidad:** Utiliza los fundamentos matemáticos (series, sumatorias, logaritmos, límites, derivadas), notaciones asintóticas, métodos para el análisis de algoritmos y medir la eficiencia de los algoritmos, tanto estructurados como recursivos, con actitud innovadora y responsable. | | | | |
| **Sem** | **Contenido** | **Actividades** | **Recursos** | **Estrategias** |
| 1 | Presentación del silabo. Introducción al curso.  Fundamentos matemáticos (Series, Sumatorias, Logaritmos, Límites y Derivadas). Algoritmos (Formalismo y Abstracción, Especificación, Terna de Hoare). Definición de complejidad algorítmica.  Resolución de ejercicios. | Socialización del silabo,  exposición del contenido, sugerencias.  Revisión de fundamentos matemáticos, complejidad, ejercicios. | Sílabo.  Comunicados, tareas.  Ejercicios:  Fundamentos matemáticos.  Lecturas:  [Cormen, 2022: introducc., cap. 1,]  [Ziviani, 20007: introducc., cap. 1,] alusivos al tema, extraer resúmenes, cuestiones en el cuaderno. | Revisión documental.  Revisión de contenidos y presentación.  Exposición.  Síntesis/retroalimentación.  Formulación de preguntas |
| 2 | Notaciones asintóticas (O, Ω, θ). Análisis de estructuras de control. Análisis de las estructuras de datos básicas. Resolución de ejercicios. Planteamiento del Proyecto.  Laboratorio: Implementación de algoritmos y comparación de su complejidad. | Evaluación de saberes previos.  Exposición del análisis básico de algoritmos, motivando la participación.  Ejemplo de ejercicio(s) para obtener la complejidad. | Comunicados, tareas.  Presentación y ejercicios de Complejidad algorítmica.  Lecturas:  [Cormen, 2022: cap. 1,2]  [Kleinberg, J., & Tardos, E., 2014.] alusivos al tema, extraer resúmenes, cuestiones en el cuaderno. | Exposición.  Síntesis/retroalimentación.  Formulación de preguntas |
| 3 | Algoritmos Recursivos  - Análisis utilizando métodos:  Substitución, Árbol de recursión, Maestro.  Resolución de ejercicios.  Laboratorio: Implementación de  algoritmos iterativos y recursivos.  Comparación de su complejidad. | Evaluación de saberes previos.  Exposición del análisis de algoritmos recursivos, desarrollando la participación.  Ejemplo de ejercicio(s) para obtener la complejidad. | Comunicados, tareas.  PPT, ejercicios: Algoritmos recursivos, recurrencias.  Lecturas:  [Cormen, 2022: cap. 1,2,3] alusivos al tema, extraer resúmenes, cuestiones en el cuaderno. | Exposición.  Síntesis/retroalimentación.  Formulación de preguntas. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Unidad II: Análisis de algoritmos de: ordenación, búsqueda y grafos. Dispersión.** | | | | |
| **Capacidad:** Aplica análisis de complejidad en algoritmos de ordenación, búsqueda, grafos mediante algoritmos para ordenar, buscar, algoritmos en grafos, algoritmos de búsqueda por dispersión, con actitud ética y responsable. | | | | |
| **Sem** | **Contenido** | **Actividades** | **Recursos** | **Estrategias** |
| 4 | Algoritmos de Ordenación  - Análisis de complejidad de los algoritmos comunes  (Bubble Sort, Insertion Sort, Selection Sort, Merge Sort,  Shell Sort, Quick Sort,  HeapSort)  - Análisis de Complejidad de los algoritmos especiales  Tiempo Lineal (Radix Sort, Counting Sort)  Ejemplos de aplicación y  Resolución de ejercicios  Laboratorio: Implementación de algoritmos de ordenación.  Comparación de su complejidad | Evaluación de saberes previos.  Exposición del análisis de algoritmos para ordenar, haciendo efectiva la participación.  Ejemplo de ejercicio(s) para obtener la complejidad de métodos para ordenar. | Comunicados, tareas.  PPT, ejercicios algoritmos de ordenación análisis de complejidad.  Lecturas:  [Kleinberg, J., & Tardos, E., 2014] alusivos al tema, extraer resúmenes, cuestiones en el cuaderno. | Exposición.  Síntesis/retroalimentación.  Formulación de preguntas |
| 5 | Análisis de algoritmos de Búsqueda y Selección  - Análisis de Complejidad  (secuencial, por bloques-Quick select, binaria, indexada, árbol binario de búsqueda-Heaps)  Ejemplos de aplicación y  resolución de ejercicios  Laboratorio: Implementación de algoritmos de búsqueda.  Comparación de su complejidad. | Evaluación de saberes previos.  Exposición del análisis de algoritmos para ordenar, logrando la participación.  Ejemplo de ejercicio(s) para obtener la complejidad en búsqueda y selección. | Comunicados, tareas.  PPT, ejercicios de algoritmos de búsqueda y selección análisis de complejidad.  Lecturas:  [Kleinberg, J., & Tardos, E., 2014] alusivos al tema, extraer resúmenes, cuestiones en el cuaderno. | Exposición.  Síntesis/retroalimentación.  Formulación de preguntas |
| 6 | Análisis de Algoritmos en Grafos - Algoritmos de Búsqueda en Anchura, Búsqueda en Profundidad, Orden Topológico, Componentes fuertemente Conexas.  Ejemplos de aplicación y resolución de ejercicios  Laboratorio: Implementación de algoritmos de grafos. Comparación de su complejidad. | Evaluación de saberes previos.  Exposición del análisis de algoritmos en grafos, motivando la participación.  Ejemplo de ejercicio(s) para obtener la complejidad en grafos.  **PRÁCTICA CALIFICADA 01** | Comunicados, tareas.  Presentación y ejercicios de algoritmos en grafos, análisis de complejidad.  Lecturas:  [Kleinberg, J., & Tardos, E., 2014] alusivos al tema, extraer resúmenes, cuestiones en el cuaderno. | Exposición.  Síntesis/retroalimentación.  Formulación de preguntas. |
| 7 | Dispersión: dispersión cerrada, dispersión abierta. Resolución de  colisiones  - Evaluación de algoritmos.  Ejemplos de aplicación y resolución de ejercicios  Laboratorio: Implementación de algoritmos de búsqueda por dispersión. | Evaluación de saberes previos.  Exposición de dispersión, motivando la participación.  Ejemplo de dispersión abierta, cerrada, complejidad. | Comunicados, tareas.  Tablas de dispersión. | Exposición.  Síntesis/retroalimentación.  Formulación de preguntas |
| 8 | **EXAMEN PARCIAL** | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Unidad III: Técnicas de diseño de algoritmos** | | | | |
| **Capacidad:** Comprende las diversas técnicas de diseño de algoritmos, sus ventajas y desventajas, análisis de complejidad, para la solución de problemas de distinta naturaleza, principalmente los de optimización, utilizando los algoritmos voraces, división y conquista, programación dinámica, algoritmos de retroceso, problemas P, NP, con actitud de innovación y responsabilidad. | | | | |
| **Sem** | **Contenido** | **Actividades** | **Recursos** | **Estrategias** |
| 9, 10 | Algoritmos Voraces  - Problema de la asignación de Carga: Interval Scheduling  - Problema de la Mochila:  Knapsack Problem  - Problema del cambio de moneda: Coin Changing  - Algoritmos voraces en Grafos  - Caminos Cortos en Grafos: Dijkstra's Algorithm  - Puentes en grafos:  Selecting Breakpoints  - Cobertura de vértices  (Vertex cover)  - Árbol de Expansión mínimo  - Prim's Algorithm  - Kruskal's Algorithm  Laboratorio: Implementación de algoritmos voraces. | Evaluación de saberes previos.  Exposición de algoritmos voraces, motivando la participación.  Ejemplo de algoritmo voraz: cambio de moneda, problema de la mochila, algoritmo voraz en grafos, algoritmo de Dijkstra, complejidad.  **PROYECTO**  Definición, consideraciones del Proyecto. | Comunicados, tareas.  Presentación y ejercicios de algoritmos voraces, complejidad. | Exposición.  Síntesis/retroalimentación.  Formulación de preguntas |
| 11 | División y Conquista  - Análisis del Algoritmo Merge Sort.  - Problema de contar inversiones: Counting Inversions.  - Problema del par de puntos cercanos: Closest Pair of Points.  - Problema de la multiplicación de N enteros: Integer  Multiplication.  - Problema de la multiplicación de matrices: Matrix Multiplication.  Laboratorio: Implementación de algoritmos División y conquista. | Evaluación de saberes previos.  Exposición de algoritmos en división y conquista, motivando la participación.  Ejemplo de algoritmo: merge sort, contador de inversiones, par de puntos cercanos, complejidad. | Comunicados, tareas.  Presentación y ejercicios de algoritmos en división y conquista, complejidad. | Exposición.  Síntesis/retroalimentación.  Formulación de preguntas |
| 12, 13 | Programación dinámica  - Definición del Criterio óptimo (OPT).  - Problema de la asignación de Tareas ponderadas: Weighted  Interval Scheduling  - Problema de la mayor subsecuencias creciente: Longest Increasing Subsequence  - Problema de la mochila óptima: Knapsack Problem  - Problema de la alineación de secuencia: Sequence Alignment.  - Problema de la alineación de secuencia en espacio Lineal: Sequence Alignment in Linear Space.  Laboratorio: Implementación de algoritmos de Programación dinámica | Evaluación de saberes previos.  Exposición de algoritmos utilizando programación dinámica, motivando la participación.  Ejemplo de algoritmo en programación dinámica: cambio de moneda, problema de la mochila, camino de costo mínimo en un grafo, problema de la alineación de secuencia, complejidad.  Revisión de Informe del Proyecto. | Comunicados, tareas.  PPT y ejemplos de algoritmos en programación dinámica, complejidad | Exposición.  Síntesis/retroalimentación.  Formulación de preguntas |
| 14 | Algoritmos de retroceso  - Ramificación y poda  Problemas P, NP y NP-Completos. - Introducción a las clases P, NP y NP completos.  - Reducción y completitud NP: (Reducción, reducciones polinómicas, máquinas de  Turing, no determinista, teorema de Cook, completitud NP, pruebas de integridad NP, jerarquía en complejidad computacional).  - Tratamiento de problemas NP completos: (algoritmos aproximados, aseguramiento de la calidad, búsqueda heurística, algoritmos heurísticos x algoritmos exactos, enumeración.  Laboratorio: Implementación de algoritmos de retroceso | Evaluación de saberes previos.  Exposición de ramificación y poda, problemas P, NP, NP-completos, motivando la participación.  Revisión de Informe del Proyecto. | Comunicados, tareas.  PPT y ejemplos de algoritmos en programación dinámica, complejidad. | Exposición.  Síntesis/retroalimentación.  Formulación de preguntas. |
| 15 | Presentación y exposición del proyecto | Exposición del proyecto | Presentaciones, algoritmos, programas, aplicaciones, análisis de complejidad algorítmica, informe. | Exposición.  Síntesis/retroalimentación.  Formulación de preguntas. |
| 16 | **EXAMEN FINAL** | Examen Final | Examen | Formulación de preguntas. |

1. **ESTRATEGIA DIDÁCTICA**

Las estrategias para utilizar durante el desarrollo de la asignatura serán las siguientes:

* **Aprendizaje por competencias.** Enseñanza y aprendizaje centrado en el desarrollo de conocimientos y capacidades prácticas, aprendizaje activo y participativo e integración de diferentes áreas del conocimiento que se pueden utilizar a situaciones cotidianas y reales.
* **Aprendizaje activo.** Los estudiantes participan activamente en el proceso a través de ejercicios o proyectos
* **Aprendizaje colaborativo**. Los estudiantes realizan trabajos juntos o en grupos pequeños para completar tareas, problemas o proyectos.

1. **EVALUACIÓN**

La evaluación de los participantes es integral, continua y permanente. Se tomarán en cuenta los conocimientos adquiridos y la capacidad de los participantes de acuerdo con los siguientes criterios:

1. Responsabilidad del participante respecto a la asistencia obligatoria al 70% de las clases.

2. Se tendrá en cuenta la intervención activa de los participantes durante las clases, desarrollo de prácticas y laboratorios y presentación del proyecto de aplicación.

El sistema evaluativo para aprobar la asignatura exige a los alumnos una nota mínima de 11, resultante de las 3 notas indicadas por los siguientes conceptos:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Unidades**  **de**  **aprendizaje** | **Criterios y logros de**  **aprendizaje** | **Procedimientos (Productos)** | **Instrumentos de evaluación** | **Pesos en porcentaje** | |
| **Sesiones** | **Notas**  **SUM** |
| 1 y 2 | Específica, diseña y evalúa soluciones algorítmicas.  Analiza la complejidad algorítmica | Examen Parcial | Evaluación  escrita | 30% | EP |
| 1, 2 y 3 | Específica, diseña y evalúa soluciones algorítmicas.  Aplica análisis de algoritmos recursivos | Promedio de Prácticas  calificadas | Evaluación  escrita | 40% | EC |
| 1, 2 y 3 | Resuelve problemas en grupo e individual, planteando e implementado soluciones eficientes | Promedio de Prácticas de laboratorios | Laboratorio  calificado |
| 1,2 y 3 | Elabora y desarrolla un proyecto - trabajo final del curso aplicando análisis, complejidad del algoritmo o técnica utilizada e implementación | Promedio de  Proyecto | Rúbrica |
| 3 | Resuelve problemas  usando las técnicas estudiadas y analizando la complejidad | Examen final | Evaluación  escrita | 30% | EF |
| **Total** | | | | 100% |  |
| **Promedio final = EP \* 0.2 + EC \* 0.6 + EF \* 0.2** | | | | | |

Para esto:

● La nota Práctica calificada, se obtiene promediando la(s) práctica(s) calificada(s) que se tomen en cada unidad

● La nota de Práctica de laboratorio calificada se obtiene promediando la(s) práctica(s) de laboratorio calificada(s) que se tomen en cada unidad

● La nota Proyecto-Trabajo final, corresponde a un trabajo grupal, cuyo tema está relacionado al curso, esta incluye presentación de un informe monográfico, software de aplicación y exposición, la cual es evaluada según rúbrica.

1. **BIBLIOGRAFÍA**

**BÁSICA**

* Cormen, T., Leiserson, C., Rivest, R., & Stein, C. (2022). Introduction to algorithms (fourth ed.). MIT Press
* Roughgarden Tim (2021). Algoritmos Iluminados primera parte. OJ Books
* Aditya Y. Bhargava (2019). Algoritmos. Una guía ilustrada para programadores y curiosos. Anaya Multimedia
* Gómez María, Cervantes José (2014). Introducción al análisis y diseño de algoritmos. Universidad Autónoma Metropolitana.
* Kleinberg, J., & Tardos, E. (2014). Algorithm design. Pearson/Addison-Wesley.
* Guerequeta García, R., & Vallecillo Moreno, A. (2000). *Técnicas de diseño de algoritmos* (2nd ed.). Servicio de Publicaciones e Intercambio Científico de la Universidad de Málaga.
* Dasgupta, S., Papadimitriou, C., & Vazirani, U. *Algorithms*.
* ALEN WEISS, Mark “Estructura de Datos en Java”. Addison Wesley. 2003.
* BRASSARD, G. / BRATLEY, T. “Fundamentos de Algoritmia”. Prentice Hall. 2001.
* CAIRÓ, Osvaldo. “Estructuras de Datos”. McGraw-Hill. 2006.
* CORTEZ VÁSQUEZ, Augusto “Estructura de datos y algoritmos”. 2002.
* CORTEZ VÁSQUEZ, Augusto. (2013). Algorítmica, Técnicas Algorítmicas. Lima. Perú: Cepredim
* JOYANES AGUILAR, Luis “Estructura de Datos en Java”. McGraw-Hill. 2008.
* LEE, R.C.T., TSENG, S.S., CHANG, R.C., TSAI, Y.T. “Introducción al diseño y análisis de algoritmos”. McGraw Hill. 2007.
* PELÁEZ SÁNCHEZ, José I. “Análisis y diseño de algoritmos: Un enfoque teórico práctico”. Universidad de Málaga. 2006.
* DROZDEK, Adam “Estructura de Datos y Algoritmos con Java”. Editorial Thomson. México. 2007.
* ZIVIANI, Nivio “Diseño de Algoritmos con Implementaciones en C y Pascal”. Thomson Editores –Paraninfo. 2007.

**COMPLEMENTARIA**

* AHO, Alfred V. “Estructura de Datos y algoritmos” Addison Wesley. 1988.
* ALLEN WEISS, Mark “Estructura de Datos y Algoritmos”. Addison Wesley. 1995.
* JOYANES AGUILAR, Luis “Estructura de Datos – Teoría y  Libro de Problemas”. McGraw-Hill. 1999.
* KRUSE, Robert L. “Estructura de Datos y Diseño de Programas”. Prentice Hall. 1988.

LIPSCHUTZ, Seymur Ph.D. “Estructura de Datos”. Serie SCHAUM. McGraw-Hill. 1987